**CITIZEN AI PROJECT**

**1. Title**

Citizen AI: An AI-Powered City Analysis and Citizen Services Assistant

**2. Abstract**

The emergence of Artificial Intelligence (AI) in governance has opened new possibilities for enhancing public services, citizen engagement, and urban safety. Citizen AI is a prototype system that uses large language models (LLMs) and a web-based interface to support two key use cases:

1. City Analysis – Providing an AI-generated evaluation of a city’s crime index, accident rates, and overall safety.

2. Citizen Services Assistant – Acting as a government helpdesk to answer queries about policies, schemes, and civic issues.

The system leverages IBM Granite 3.2–2B Instruct, a transformer-based LLM, integrated with Gradio for user interaction. Although currently based on simulated AI-generated data, the system demonstrates a framework that can be extended to real-time datasets from government portals, traffic systems, and law enforcement agencies. This project shows how AI can empower citizens with easy access to safety and governance-related information.

**3. Introduction**

Modern cities face increasing challenges in ensuring safety, reducing accidents, and improving communication between government institutions and citizens. Traditional systems, such as government websites or call centers, often fail to provide quick, accurate, and user-friendly information.

Artificial Intelligence, particularly Natural Language Processing (NLP), provides a solution by enabling machines to understand and generate human-like text. Citizen AI leverages this capability to create a virtual civic assistant.

For citizens, it simplifies access to information.

For policymakers, it demonstrates how AI can support better communication.

For students and researchers, it showcases the integration of AI models into practical applications.

This project focuses on demonstrating how AI-driven conversational systems can be designed for social good.

**4. Objectives**

The main objectives of this project are:

1. Develop a City Analysis Tool

Generate insights on crime and accident statistics.

Provide a general safety assessment of a given city.

2. Build a Citizen Services Assistant

Answer queries related to government schemes, civic services, and public policies.

Use AI-based natural language generation to ensure responses are human-like.

3. Create a User-Friendly Interface

Build a web application using Gradio.

Provide separate tabs for City Analysis and Citizen Services.

4. Explore AI in E-Governance

Show how LLMs can assist governance.

Propose future integration with official datasets.

**5. System Architecture**

5.1 Overview

Citizen AI follows a client-server architecture:

Frontend (User Interaction): Gradio interface that takes user inputs.

Backend (Model Processing): IBM Granite model processes prompts and generates responses.

Output Layer: AI-generated analysis or information is displayed in text form.

5.2 Flowchart

User Input → Tokenization → AI Model (IBM Granite) → Text Generation → Response Display

5.3 Components

1. Tokenizer: Converts user input into numerical form.

2. Granite Model: Processes tokens and generates meaningful text output.

3. Response Decoder: Converts AI output back into natural language.

4. Gradio UI: Provides interactive web-based tabs for usage.

**6. Features**

6.1 City Analysis

Input: City name.

Output: Crime index, accident rates, safety analysis.

Use case: A person moving to a new city can get safety insights.

6.2 Citizen Services Assistant

Input: Natural language queries (e.g., What government schemes are available for students?).

Output: AI-generated responses summarizing relevant information.

Use case: Citizens receive instant answers instead of searching multiple websites.

**7. Technology Stack**

7.1 Programming Language

Python 3.10+ – due to its strong AI/ML ecosystem.

7.2 Frameworks & Libraries

Transformers (Hugging Face) – for loading IBM Granite model.

Torch (PyTorch) – backend for deep learning computations.

Gradio – for interactive, web-based front-end.

7.3 Model Used

IBM Granite 3.2–2B Instruct

Type: Causal Language Model.

Parameters: ~2 billion.

Designed for instruction following and conversational tasks.

**8. Code Explanation**

8.1 Model Initialization

model\_name = "ibm-granite/granite-3.2-2b-instruct"

tokenizer = AutoTokenizer.from\_pretrained(model\_name)

model = AutoModelForCausalLM.from\_pretrained(model\_name)

Loads tokenizer and model. Ensures pad\_token is set.

8.2 Response Generation

def generate\_response(prompt, max\_length=1024):

inputs = tokenizer(prompt, return\_tensors="pt")

outputs = model.generate(\*\*inputs, max\_length=max\_length, temperature=0.7)

return tokenizer.decode(outputs[0], skip\_special\_tokens=True)

Encodes prompt → runs model → decodes response.

8.3 City Analysis Function

def city\_analysis(city\_name):

prompt = f"Provide detailed analysis of {city\_name}..."

return generate\_response(prompt)

8.4 Citizen Interaction Function

def citizen\_interaction(query):

prompt = f"As a government assistant, respond to: {query}"

return generate\_response(prompt)

8.5 Gradio Interface

Two tabs: City Analysis and Citizen Services.

Each has input textbox + output textbox + button.

**9. Advantages**

1. Accessibility – Simple and web-based.

2. Cost-Effective – Uses open-source models.

3. Scalable – Can connect with real APIs.

4. AI-Powered Insights – Generates text that feels natural.

**10. Limitations**

1. No Real Datasets – Crime and accident data are AI-simulated.

2. Accuracy – Model responses may not always be factually correct.

3. Compute Cost – Requires GPU for fast inference.

4. Language Limitation – Currently works in English.

**11. Future Enhancements**

API Integration with government databases (crime records, transport safety, schemes).

Multilingual Support (Tamil, Hindi, French, etc.).

Mobile App Deployment with offline mode.

Voice Assistant Feature for accessibility.

Data Visualization (graphs, charts for crime/accident statistics).

**12. Conclusion**

Citizen AI proves the potential of AI in governance. While it currently generates simulated analysis, the framework is flexible enough to connect with real-time government data, creating a reliable digital citizen assistant. With improvements, Citizen AI could evolve into a national e-governance platform that provides accessible, AI-driven civic services to millions of people.

**13. References**

1. Hugging Face – IBM Granite Models

2. Gradio Documentation – https://www.gradio.app

3. PyTorch Documentation – https://pytorch.org/docs

4. Research Papers on AI in Governance – IEEE & Springer (2022–2024)